Forward jet resolution study Aug 2, fsPHENIX biweekly meeting

- **Issues in last update (May 3)**
  - Only pythia6 tried
  - Too high min parton pT applied in cfg (ckin3 50)
  - No triggers applied / Used RMS values instead of fit in getting resolution
- **Updates in this report**
  - Pythia8: minimum energy trigger added
  - Pythia6: minimum energy, minimum jet pT, and η (almost same to PHPy8JetTrigger) triggers added
    → results look weird. I need hint/lead
- **What I'm going to talk:**
  - Reminder (definitions, cuts, procedures)
  - Resolution distributions using Pythia8 w/ triggers
  - Pythia6 results: items I checked

Reminder

- **Definitions**
  - Energy resolution: e_reco / e_true (previously: (e_true – e_reco)/e_true)
  - ϕ resolution: spread of Δϕ (ϕ_true – ϕ_reco)
  - η resolution: spread of Δη (η_true – η_reco)
- **Cuts:**
  - True η windows: {1.2, 1.6, 2.0, 2.4, 2.8, 3.2, 3.6, 4.0} (previously: {2.3, 2.9, 3.5, 4.1})
  - True energy windows: {20, 22, 24, 26, 28, 30, 34, 38, 42, 46, 50, 60, 70, 100}
- **Procedures:**
  1. Generate events using Pythia8 (Pythia6) + Geant4 simulation + Jet evaluator
     (used default setup in *Fun4All_G4_fsPHENIX.C*, framework build pulled in July 21)
  2. Search 1st/2nd leading jets (highest/2nd highest pT) in an event
  3. Fill target resolution parameter (ex. Δϕ) for given condition into a TH1
  4. Get resolution: tested RMS, Gaussian fit, Gaussian fit w/ minimum # of entries in TH1

Pythia8 conditions

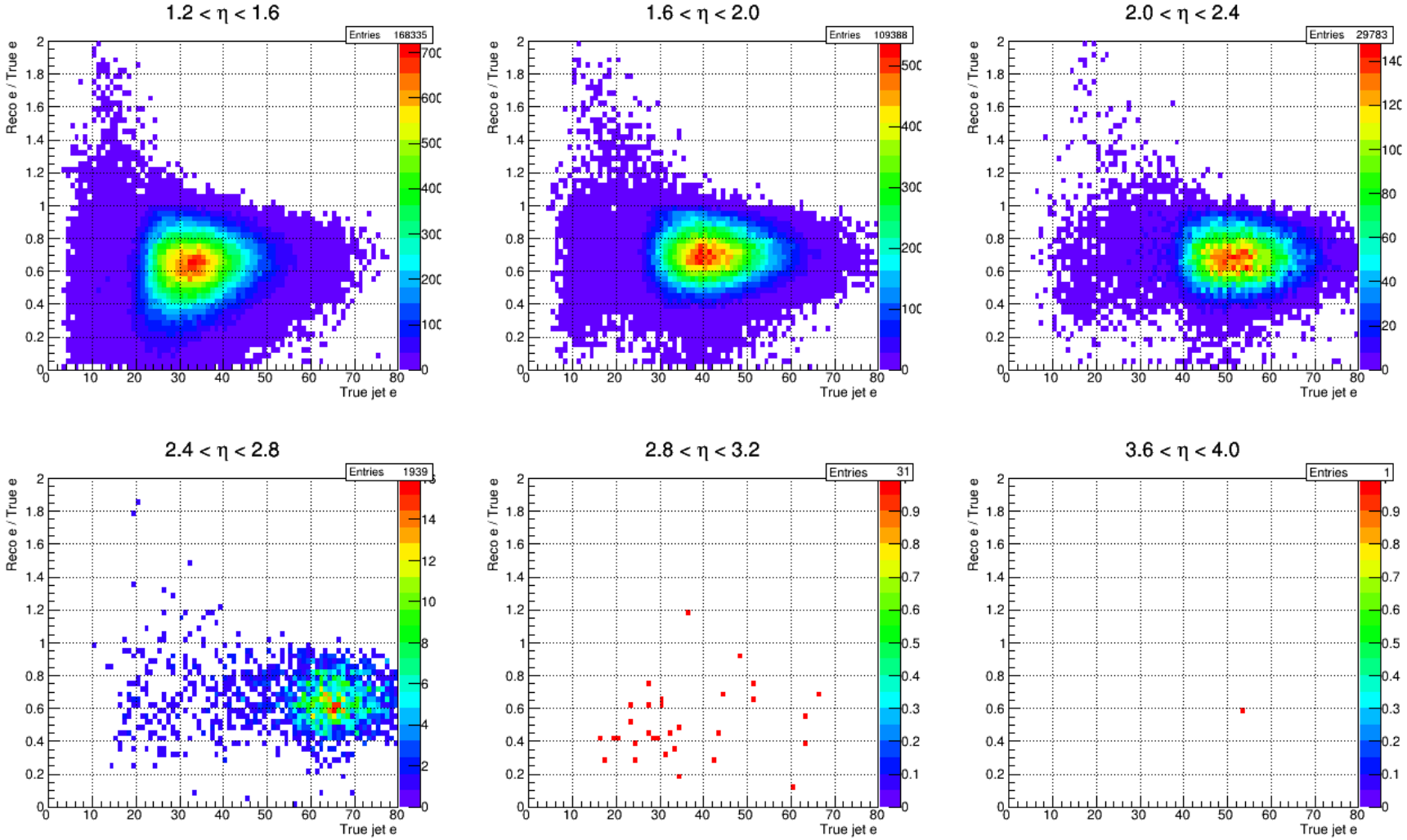
- **phpythia8.cfg**

```
! Beam settings
Beams:idA = 2212   ! first beam,  p = 2212, pbar = -2212
Beams:idB = 2212   ! second beam, p = 2212, pbar = -2212
Beams:eCM = 200.   ! CM energy of collision

! Settings related to output in init(), next() and stat()
Init:showChangedSettings = on
#Next:numberCount = 0       ! print message every n events
Next:numberShowInfo = 0          ! print event information n times
#Next:numberShowProcess = 1       ! print process record n times
#Next:numberShowEvent = 1         ! print event record n times

! PDF
#PDF:useLHAPDF = on
#PDF:LHAPDFset = CT10.LHgrid
#PDF:pSet = 7 ! CTEQ6L

! Process
#HardQCD:hardccbar = on
#HardQCD:hardbbbar = on
HardQCD:all = on
#Charmonium:all = on
#SoftQCD:nonDiffractive = on

! Cuts
PhaseSpace:pTHatMin = 5.0 (10.0, and 15.0)
```

- **Triggers**
  - $1.2 < \eta < 4.0$
  - Minimum jet energy > 10
- **Generated:**
  - Produced 0.1 M for each pTHatMin (0.1 M x 3)

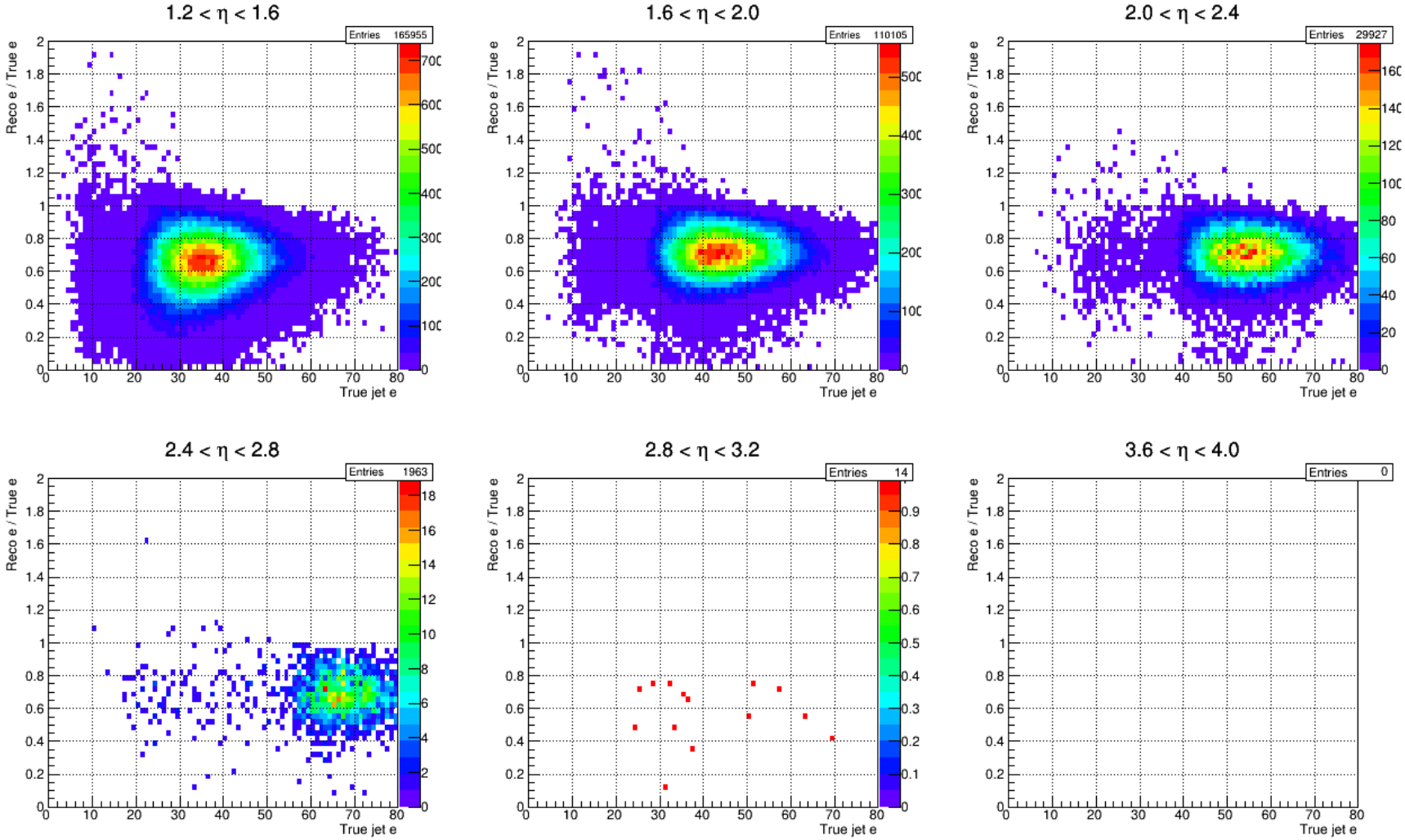
    → pTHatMin used: 5, 10, and 15
- **Evaluated:**
  - R = 0.4, 0.6

Pythia8 sanity check R = 0.4



– Reco e / True e vs. True e, 0.3 M generated, 1st/2nd leading jets only

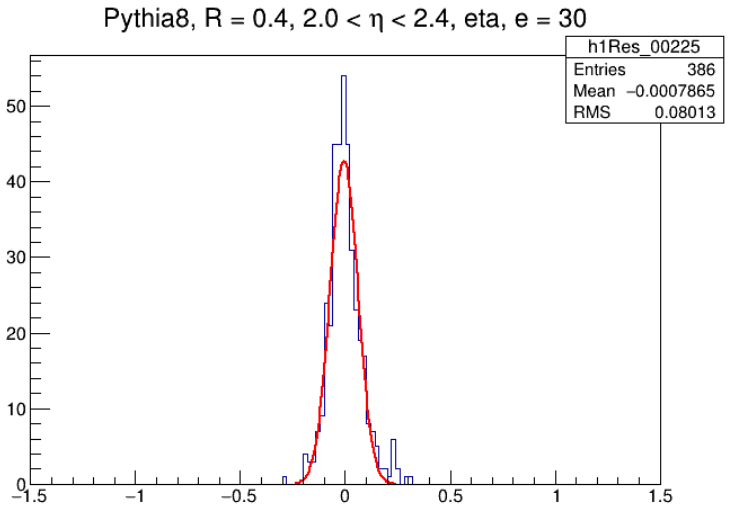
Pythia8 sanity check R = 0.6



– Reco e / True e vs. True e, 0.3 M generated, 1st/2nd leading jets only

Pythia8 jet resolution evaluation RMS, Fit, Fit + min event

- **I fill a TH1 for given condition: for instance,**
  - Pythia8, R = 0.4, 2.0 < η_true < 2.4, probe Δη, 30 < energy_true < 34 ↓



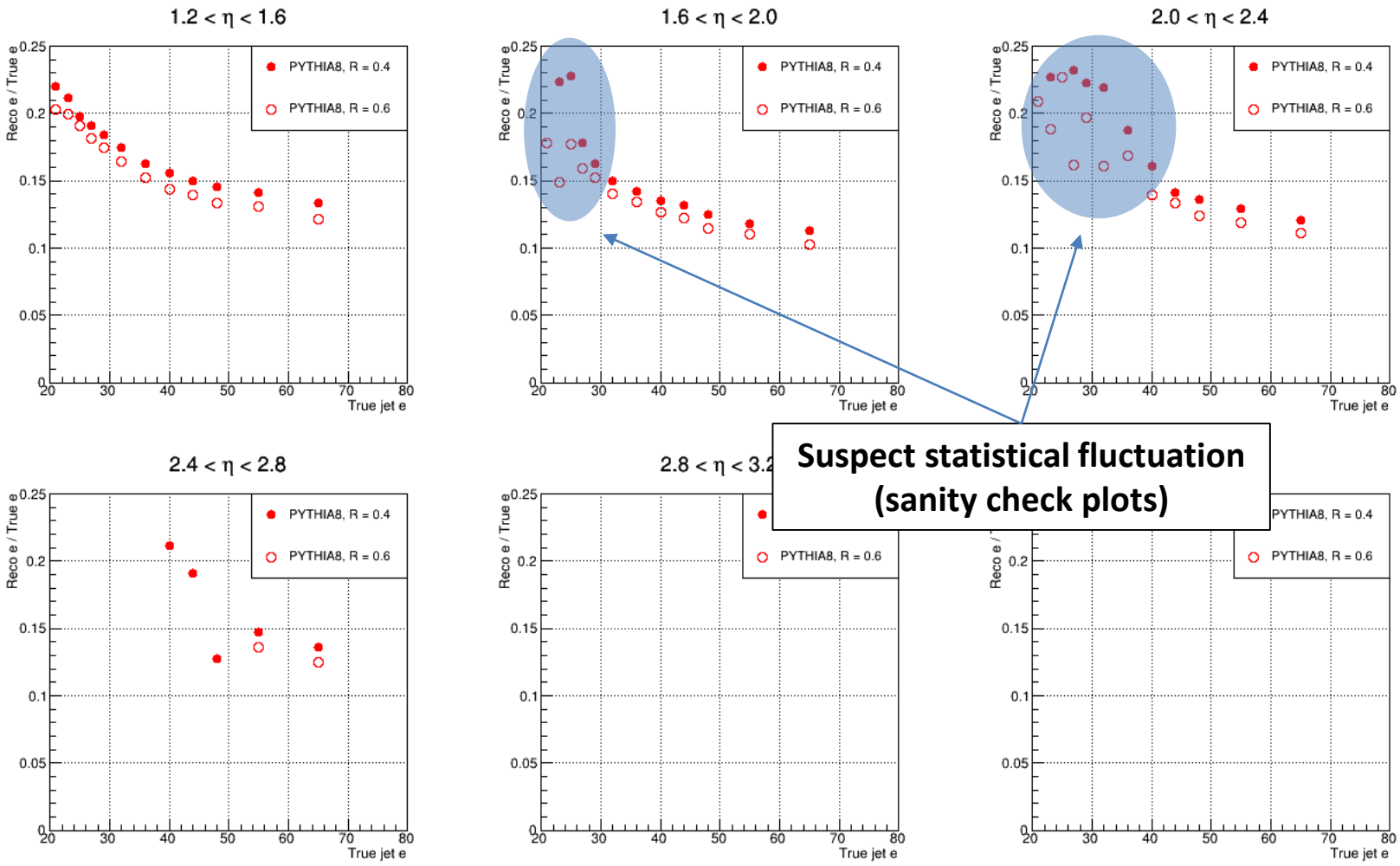
  - RMS means I directly use RMS of histogram as Δη
  - Fit means I apply fit on histogram for given condition:

    → use histogram's mean and RMS as Gaussian fit's seeds

    → Perform fit on the range of: Mean ± 3 x RMS
  - Fit + min events # means I only try previous fit when histogram's # of entries > 30

Pythia8 jet resolution evaluation

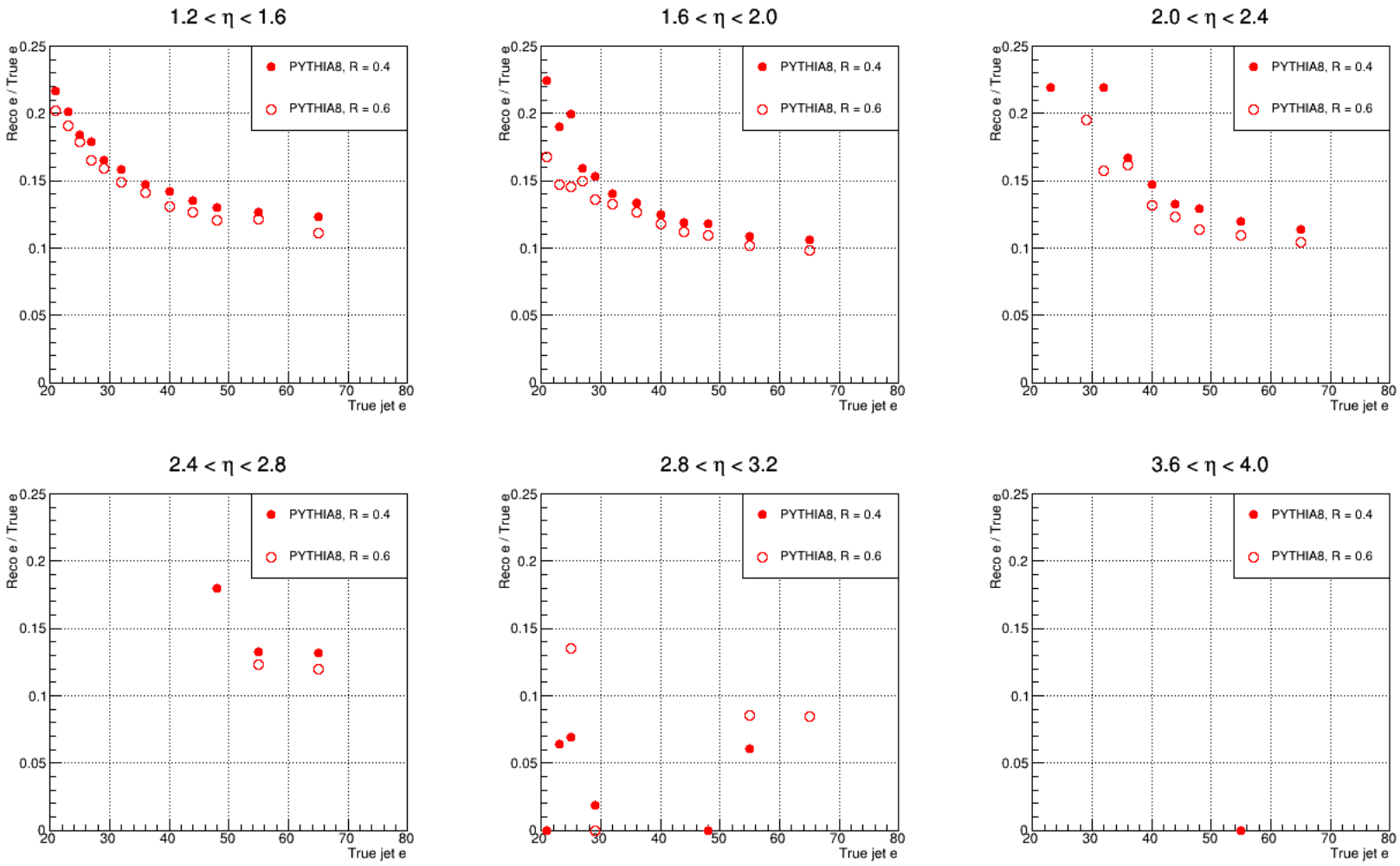
Energy, RMS



Reco E / True E vs. True E **(* Caveat: x axis begins from 20)**

Pythia8 jet resolution evaluation

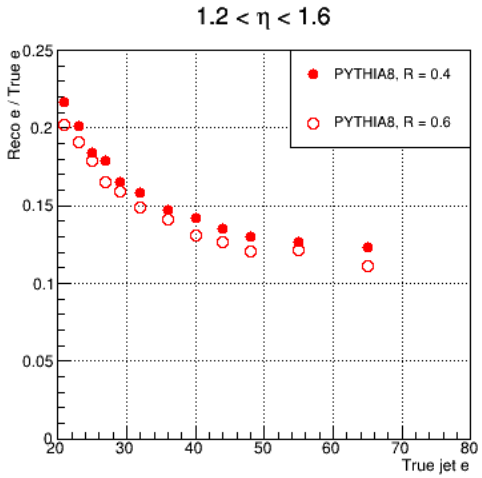
Energy, Fit

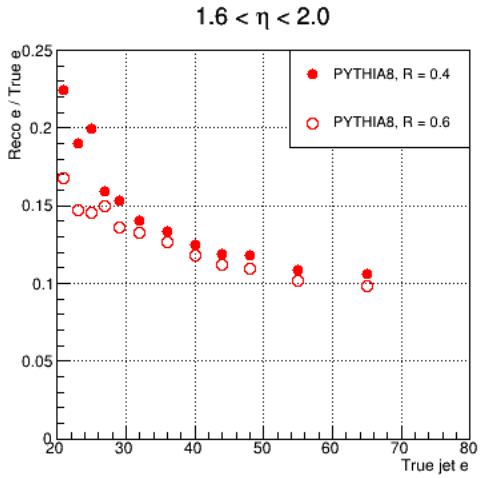


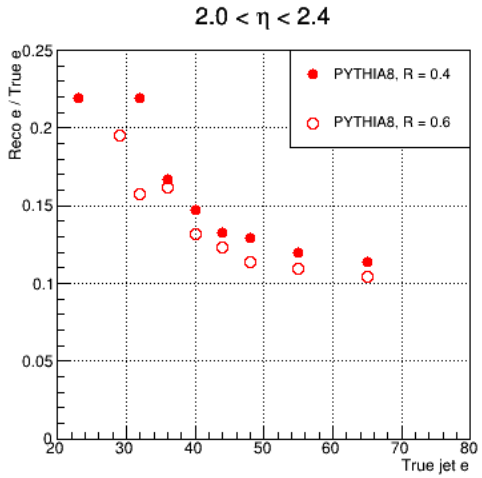
Reco E / True E vs. True E

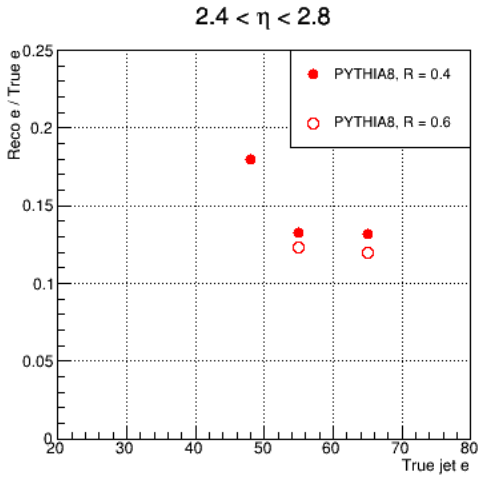
Pythia8 jet resolution evaluation

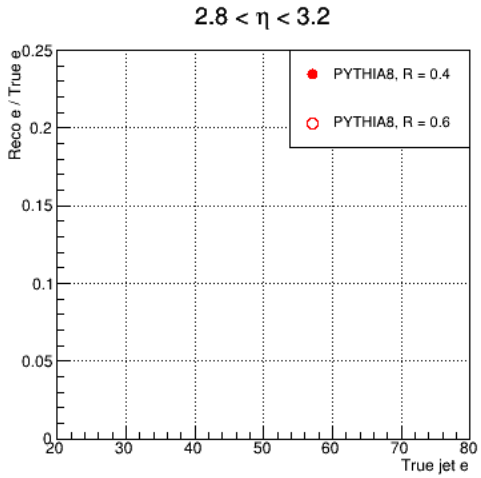
Energy, Fit + min event #

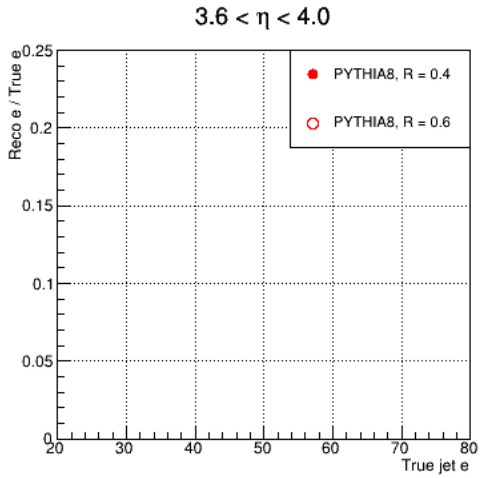






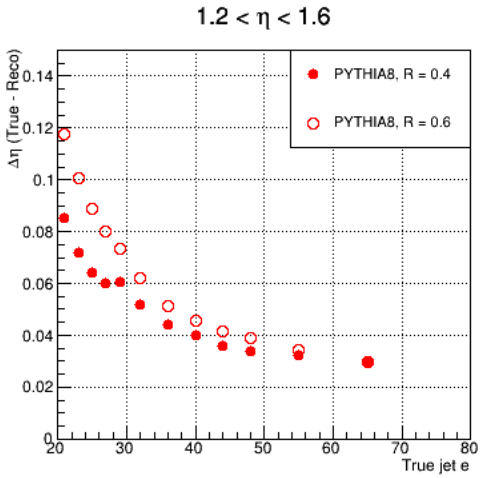


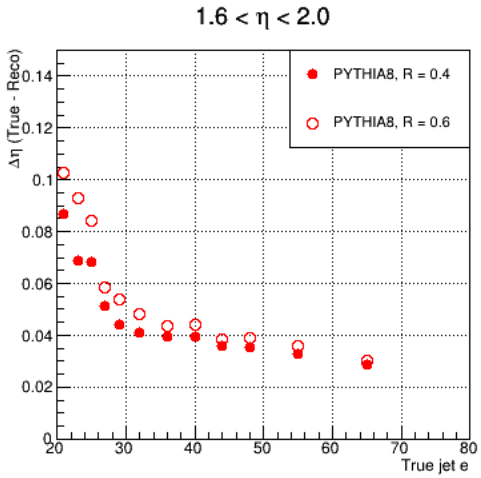


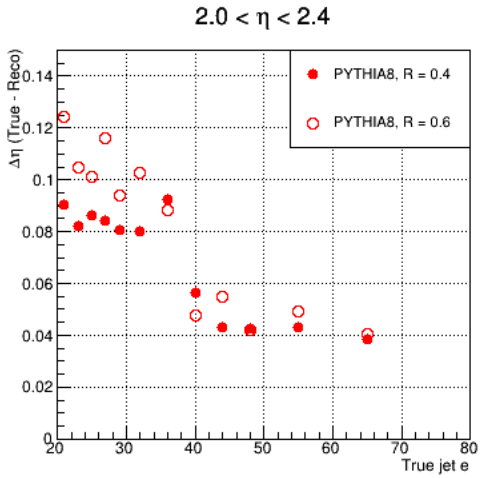


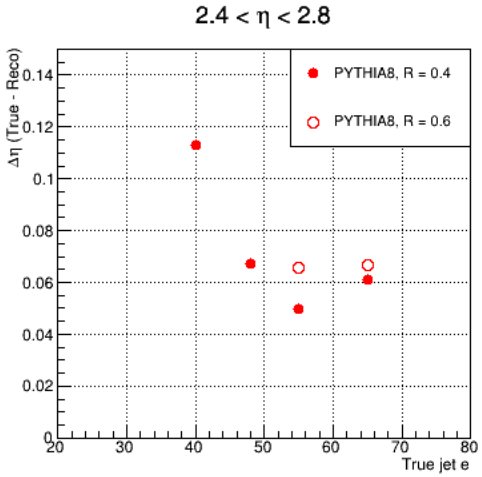
Reco E / True E vs. True E

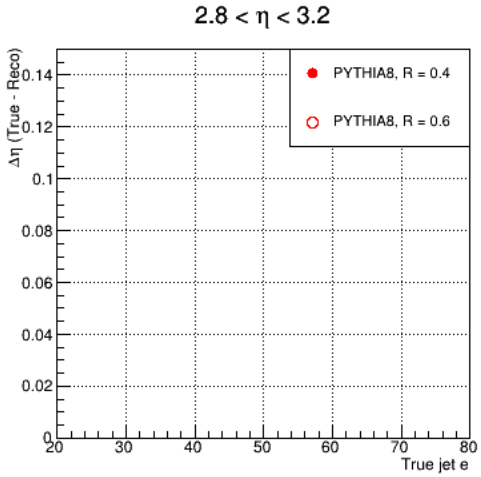
Pythia8 jet resolution evaluation η, RMS

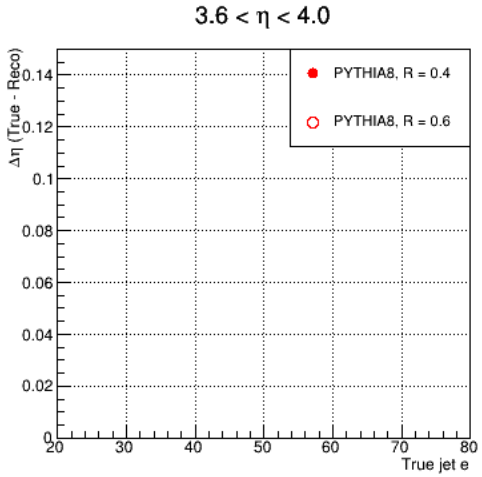






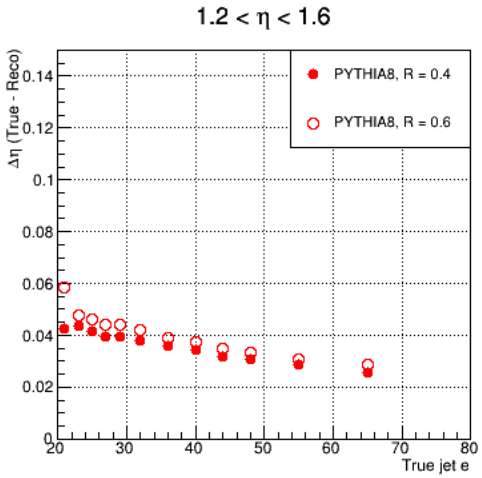


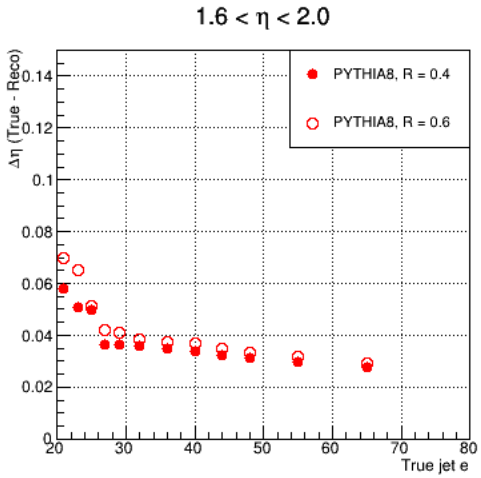


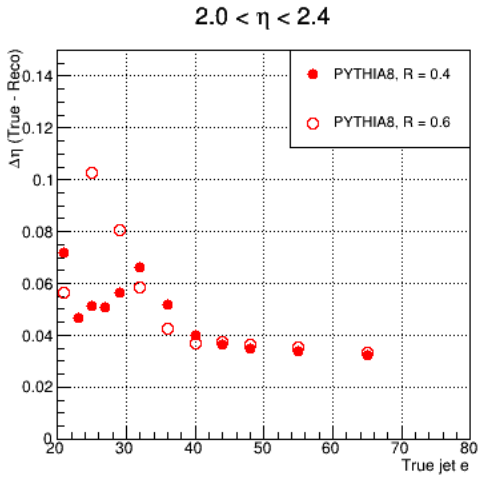


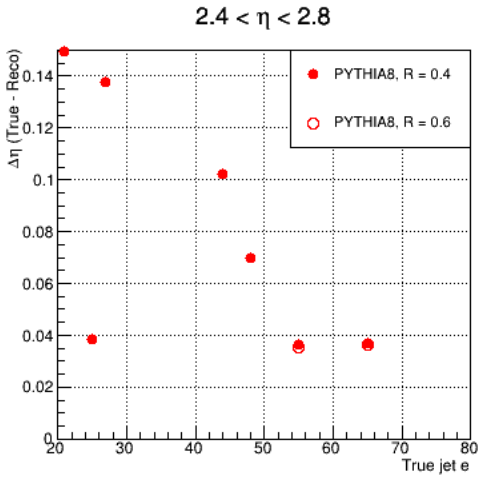
Δη vs. True E

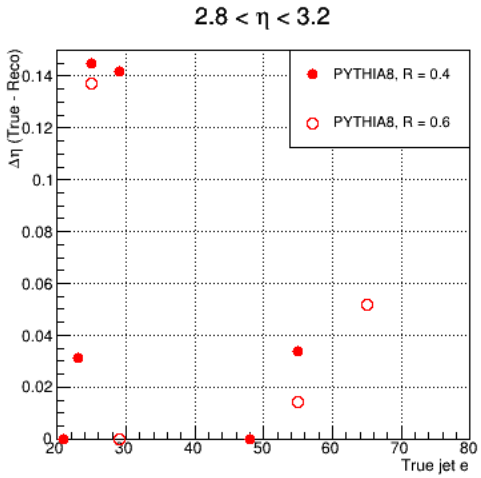
Pythia8 jet resolution evaluation η, Fit

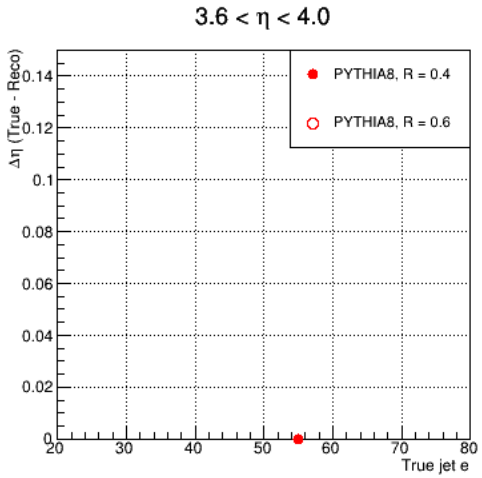








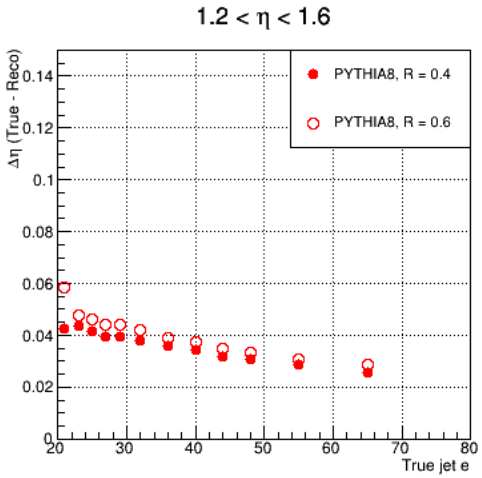


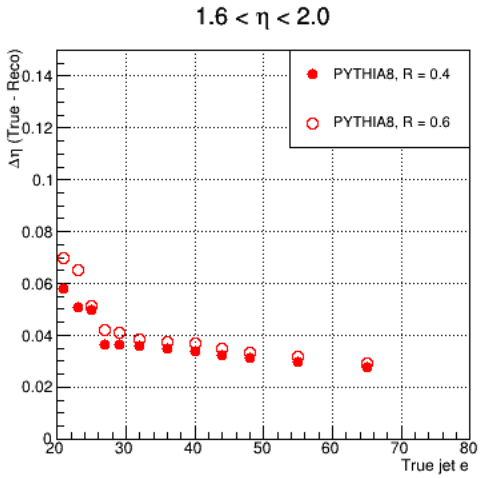


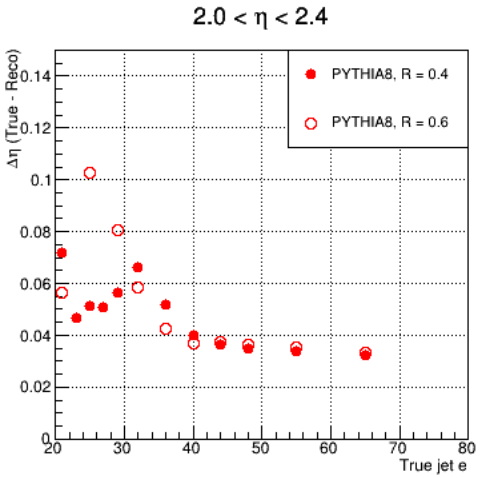
Δη vs. True E

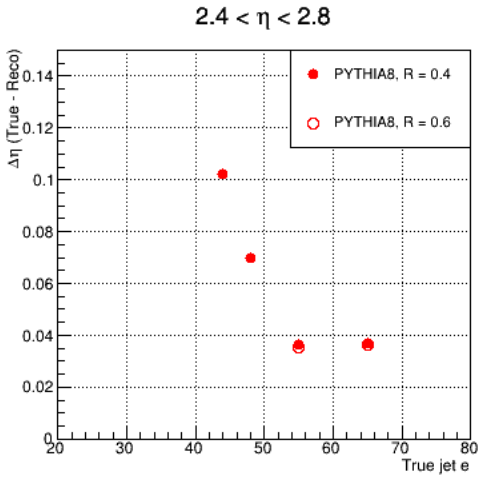
Pythia8 jet resolution evaluation

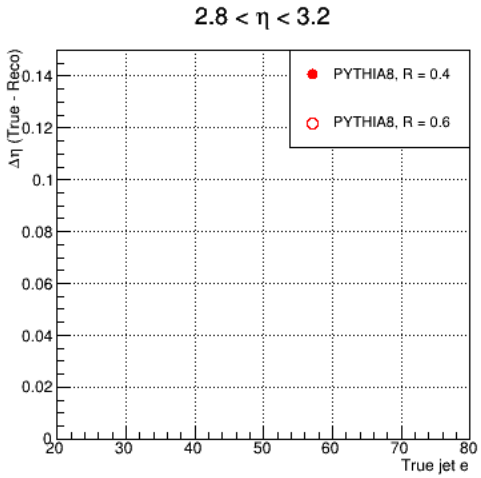
η, Fit + min event #

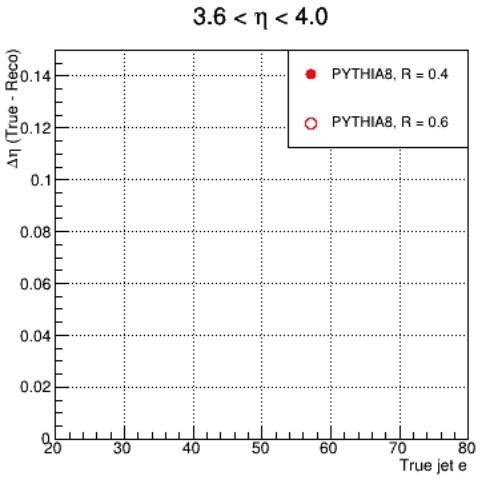






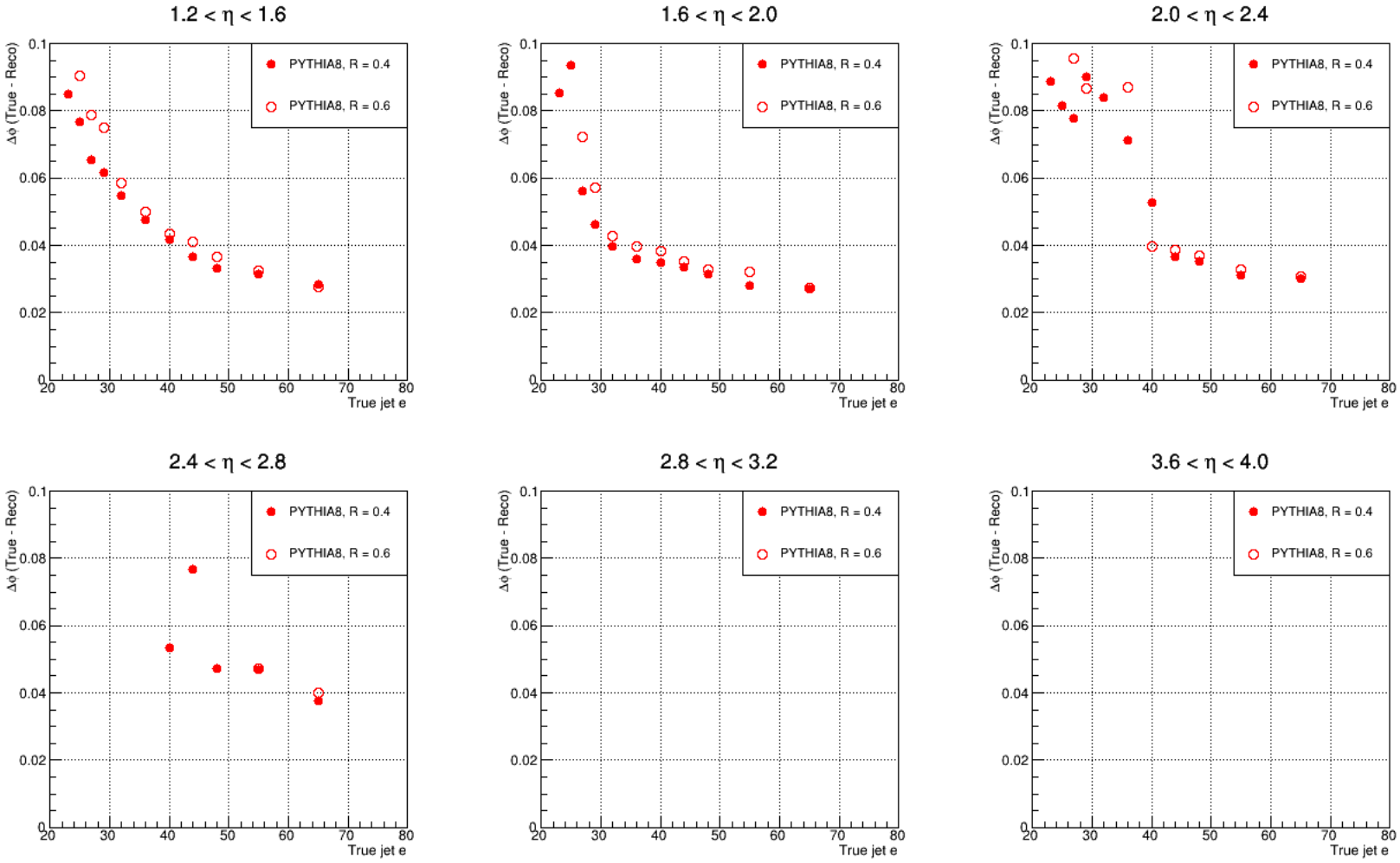






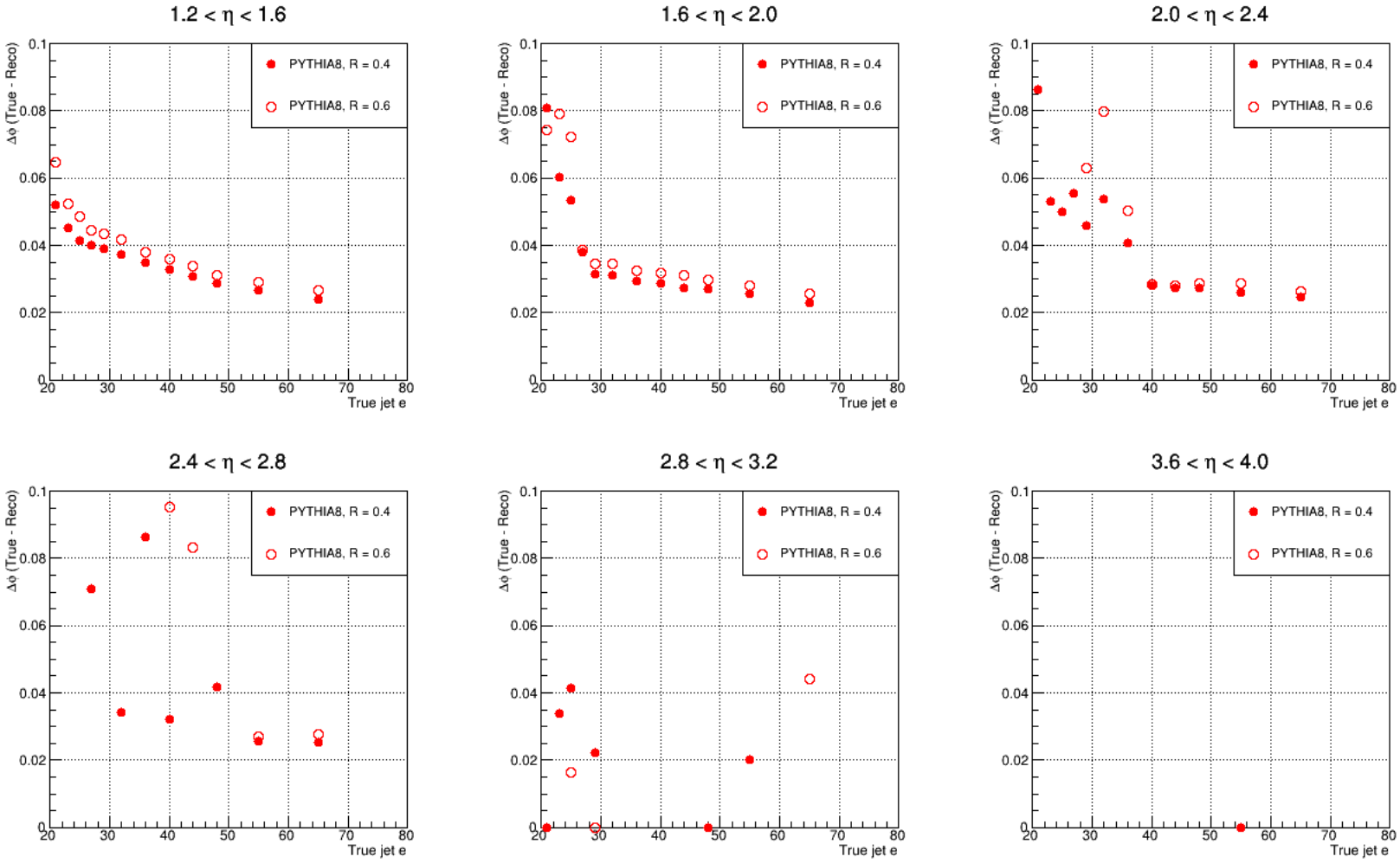
Δη vs. True E

Pythia8 jet resolution evaluation ϕ, RMS



Δϕ vs. True E

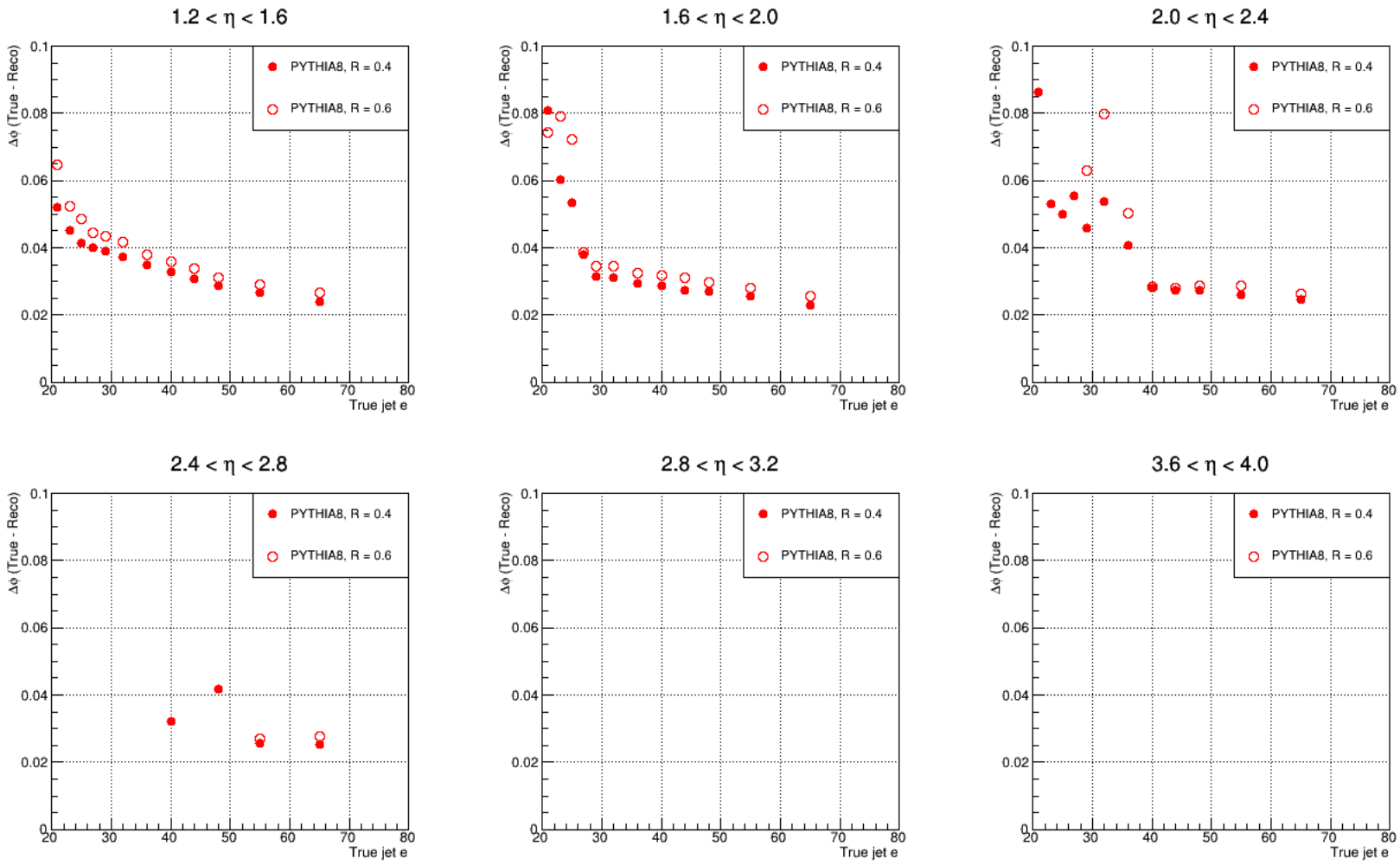
Pythia8 jet resolution evaluation ϕ, Fit



Δϕ vs. True E

Pythia8 jet resolution evaluation

ϕ, Fit + min event #



Δϕ vs. True E

Intermediate summary Pythia8

- **Pythia8 resolution plots doesn't look that bad**
  - It seems Fit + min events # makes most sense
  - I expect additional events generation will make the distribution smoother (for fluctuating points)
  - Dividing η segments above 2.4 won't be much meaningful (?)

    → a question: reason for 'not many' events? Inherently low statistics or something else?

Adding triggers on Pythia6

- **Added jet triggers**
  - If my understanding is correct, PHPy8JetTrigger works like this:

    a. access on produced pythia8 event

    b. iterate all produced particles (or sub-events?), then save a particle in the form of pseudojet (*fastjet::Pseudojet*) if it satisfy given triggers

    c. Apply jetFinder algorithm, return 'true' if min pT condition is satisfied

  - Added triggers on PHPythia6: basically it's a copy & paste of:

    1. a. part: *PHPy6ForwardElectronTrig*

       accessed *HepMC::GenEvent* instead of *pythia6*

    2. b. and c. part: *PHPy8JetTrigger* (triggering parts)

    * Snippet of the code is in backup

Pythia6 conditions

- **phpythia6.cfg**

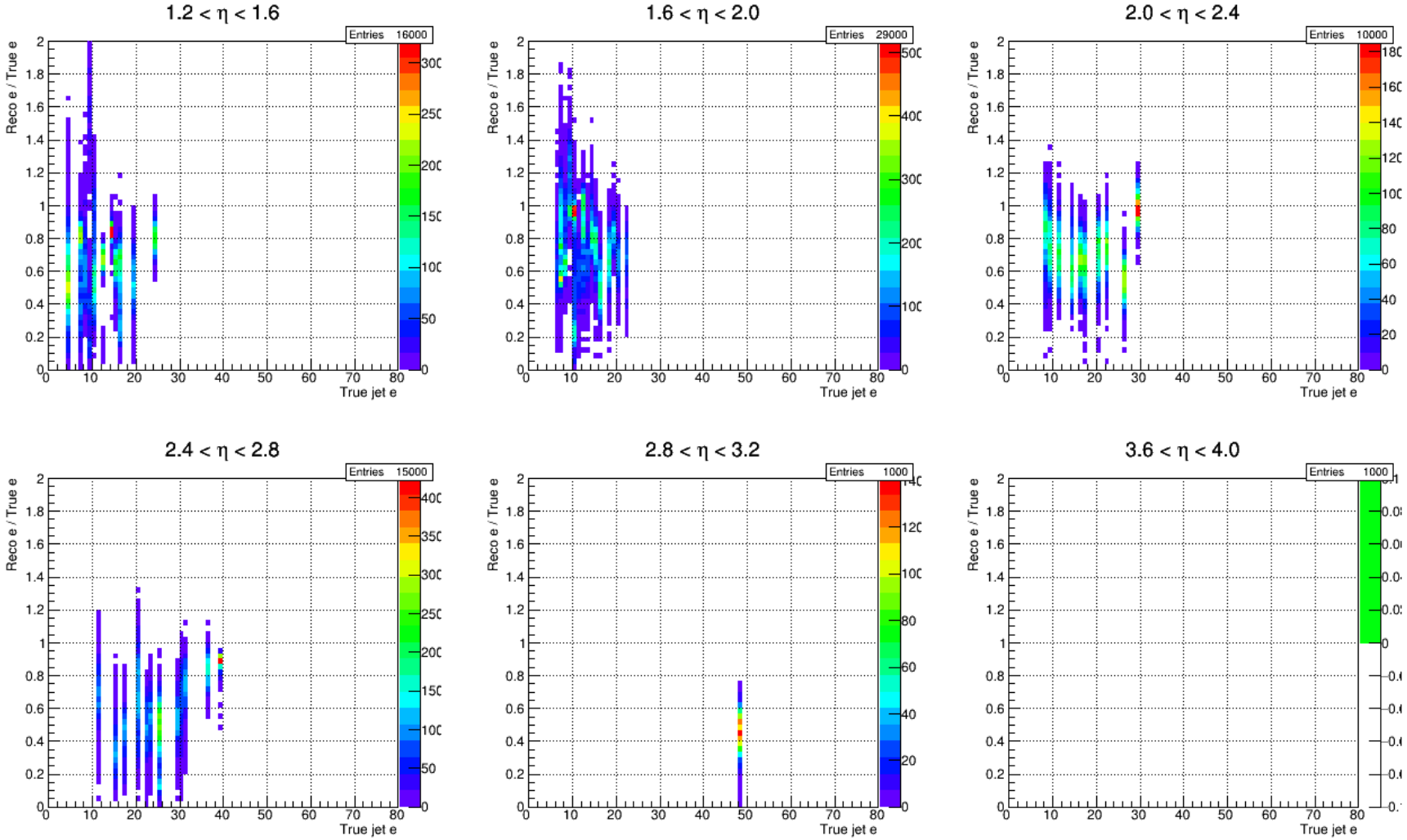
```
roots  200
proj   p
targ   p
frame  cms
pytune 100 // tune A
msel 0        // turn on all prod. mechanisms manually
msub 10 1   // f + f' -> f + f' (QFD)
msub 11 1   // f + f' -> f + f' (QCD) : QCD jets (semi-hard QCD 2 -> 2 processes)
msub 12 1   // f + fbar -> f' + fbar'
msub 13 1   // f + fbar -> g + g
msub 28 1   // f + g -> f + g
msub 53 1   // g + g -> f + fbar
msub 68 1   // g + g -> g + g
msub 83 1   // f + q -> f' + Q, massive
ckin 3 5.0  // min parton pt  10.0, and 15.0 as well
```

- **Triggers**
  - $1.2 < \eta < 4.0$
  - Minimum jet energy > 10
- **pytune 100**
  - Added a couple of lines on PHPythia6 to use it

  * It was omitted in recent build. Why?
- **Generated:**
  - Produced 0.1 M for each ckin3 (0.1 M x 3)

    → ckin5 used: 5, 10, and 15
- **Evaluated:**
  - R = 0.4, 0.6

Pythia6 sanity check R = 0.4, 0.3 M events



– Yep. I know. Something is wrong.

Pythia6 sanity check

- **So what's wrong?**
  - Bug in the macro? Pythia8 shares same macro. Thus it's not.
  - Didn't used proper configuration?

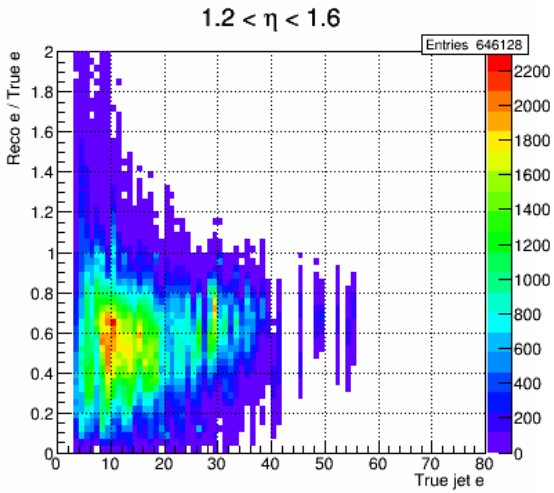
    → possibly
  - Triggers didn't worked properly?

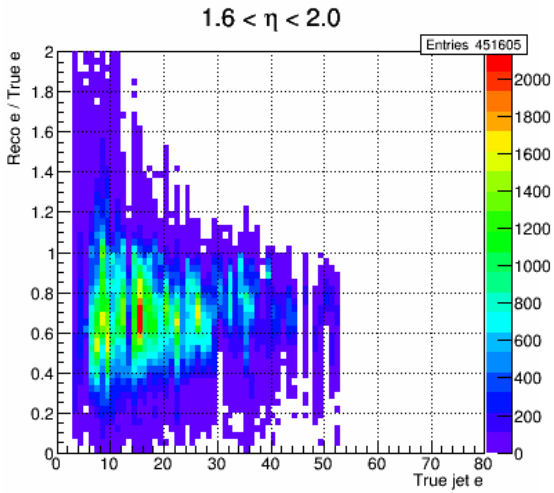
    → likely. I put snippet of my codes in backup. Any hint?
  - Pytune wasn't applied?

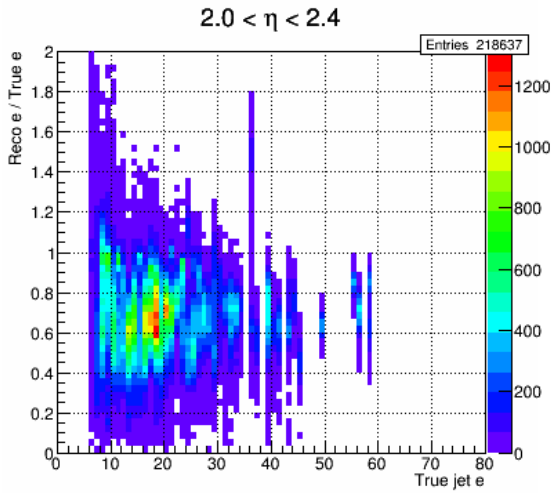
    → also likely
  - Anyway, I produced much more events (3M = 1 M x 3 ckin3 setup (5, 10, and 15))

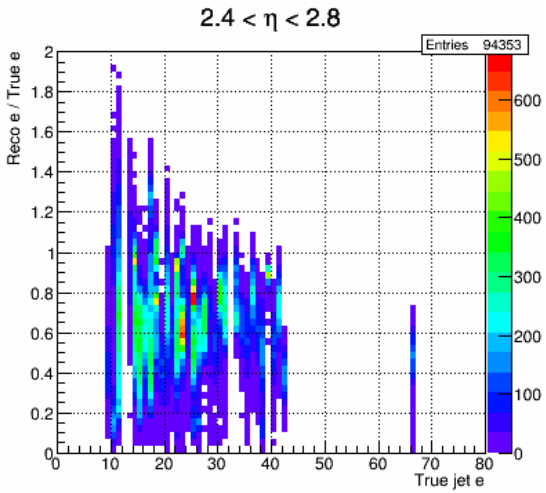
    and checked resolution

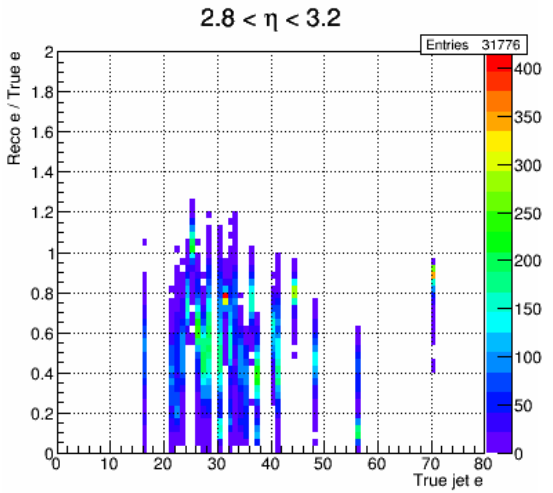
Pythia6 sanity check R = 0.4, 3 M events

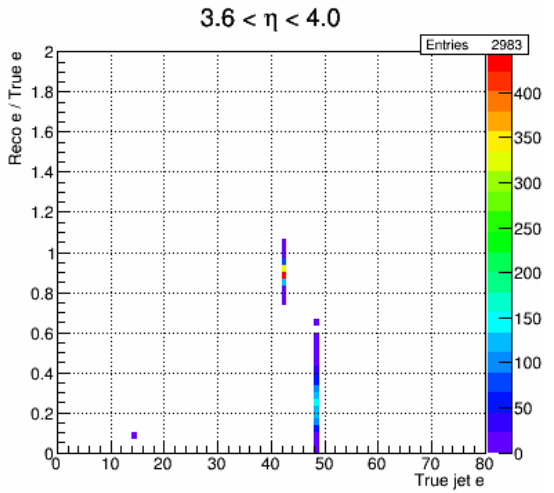




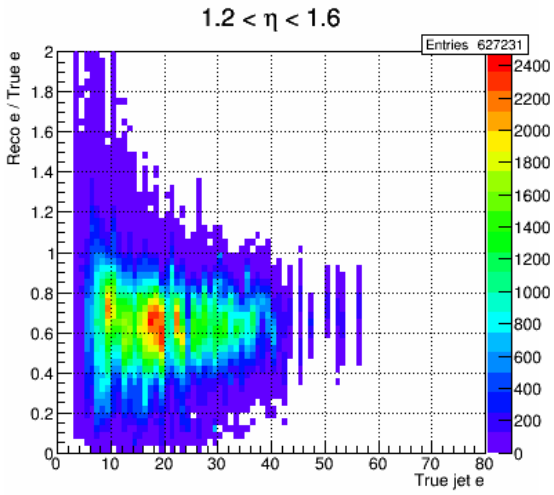


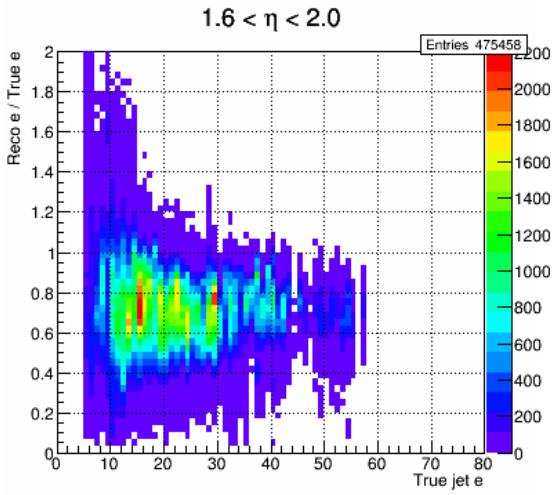


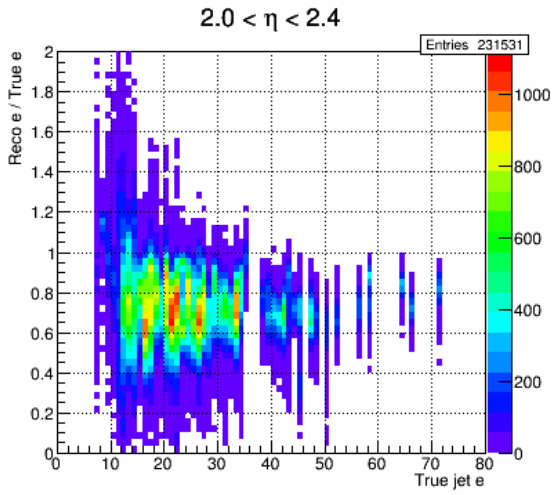


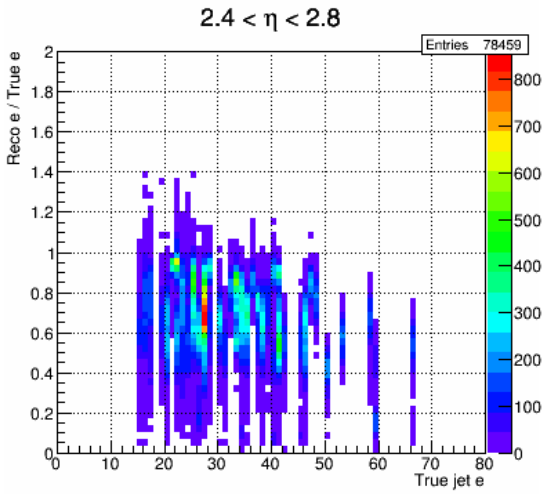


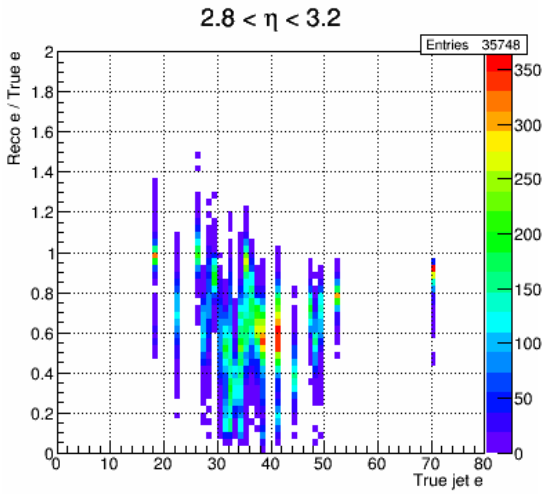
Pythia6 sanity check R = 0.6, 3 M events

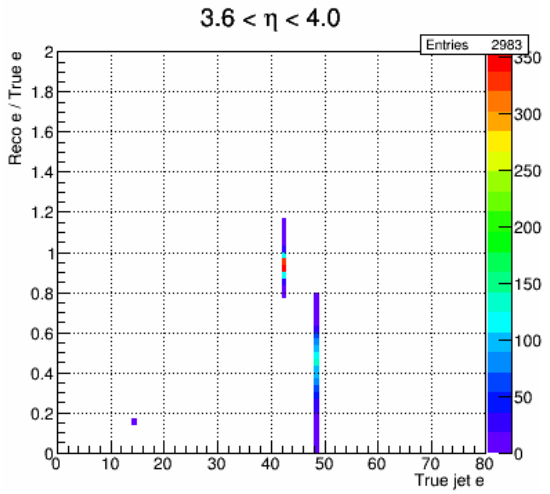






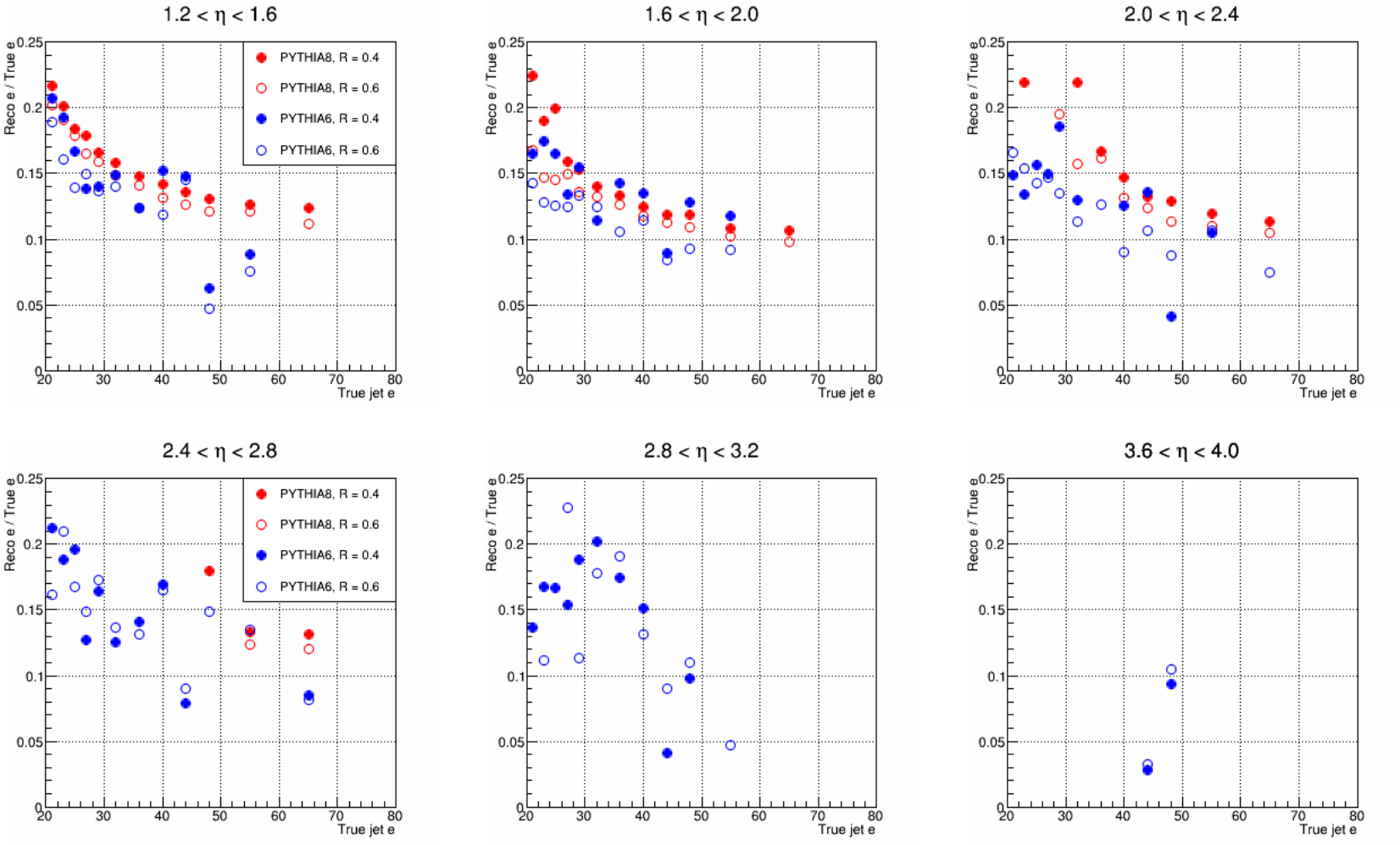






Pythia8 jet resolution evaluation

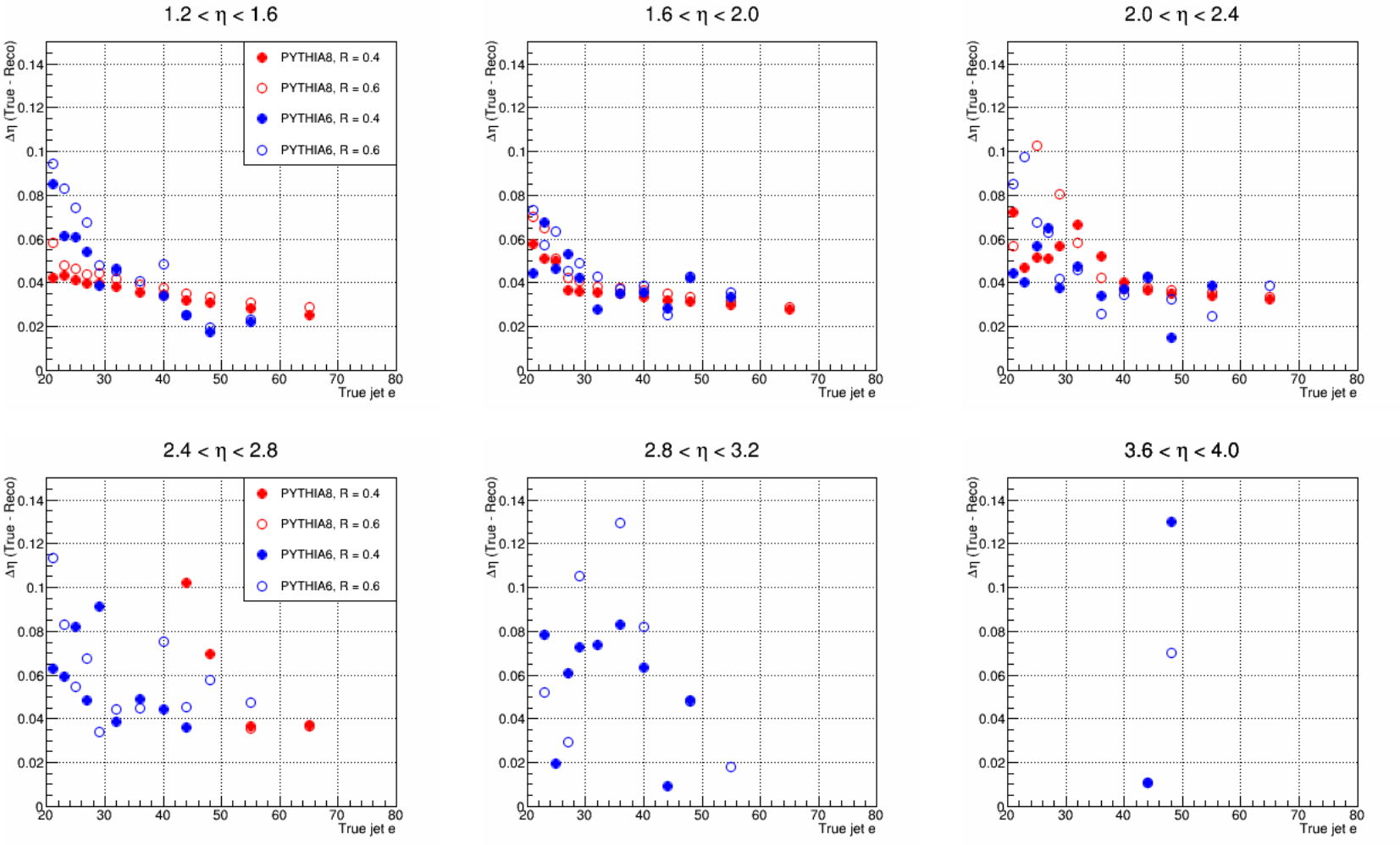
e, Fit + min event #



Reco E / True E vs. True E

Pythia8 jet resolution evaluation

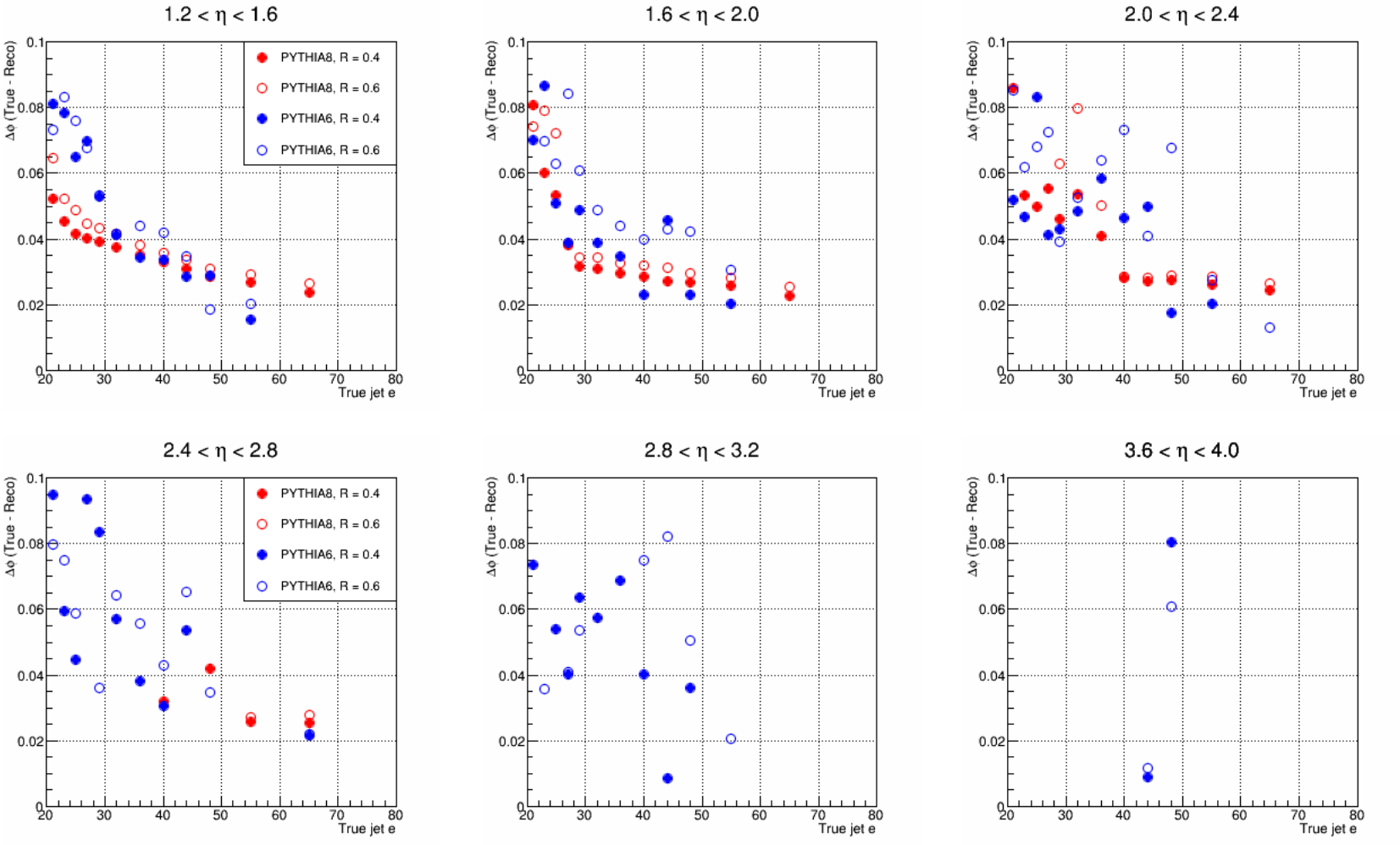
η, Fit + min event #



Δη vs. True E

Pythia8 jet resolution evaluation

ϕ, Fit + min event #



Δϕ vs. True E

Last slide

Backup Pythia8: energy trigger (newly added)

```
// Loop over all particles in the event
std::vector<fastjet::PseudoJet> pseudojets;
for (int i=0; i<pythia->event.size(); ++i)
{
    if (pythia->event[i].status() > 0) //only stable particles <- stable?
    {
        //remove some particles (muons, taus, neutrinos)...
        //12 == nu_e
        //13 == muons
        //14 == nu_mu
        //15 == taus
        //16 == nu_tau
        if ((abs(pythia->event[i].id()) >= 12) && (abs(pythia->event[i].id()) <= 16)) continue;

        //remove acceptance... _etamin, _etamax
        if ((pythia->event[i].px() == 0.0) && (pythia->event[i].py() == 0.0)) continue; //avoid pt=0
        if ((pythia->event[i].eta() < _theEtaLow) || (pythia->event[i].eta() > _theEtaHigh)) continue;
        if (pythia->event[i].e() < _minEnergy) continue; //!

        fastjet::PseudoJet pseudojet(
                pythia->event[i].px(),
                pythia->event[i].py(),
                pythia->event[i].pz(),
                pythia->event[i].e()
                );
        pseudojet.set_user_index(i);
        pseudojets.push_back(pseudojet);
    }
}
```

Backup Reference (sPHENIX proposal)

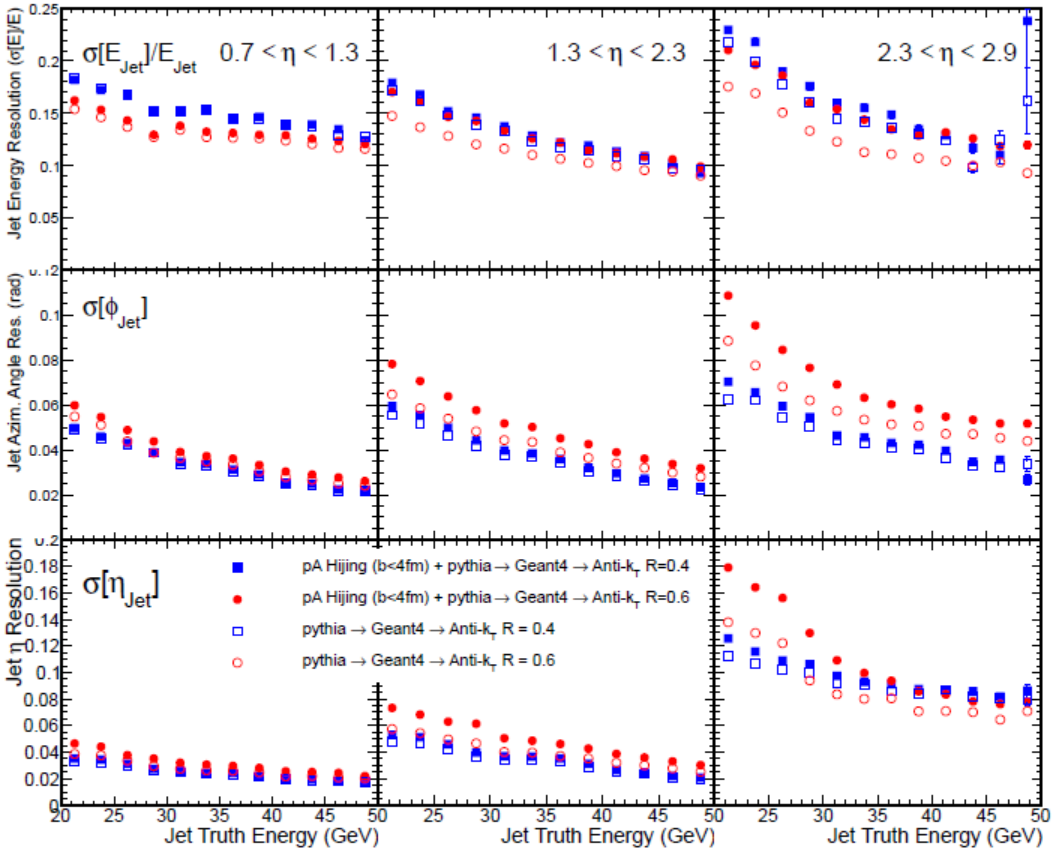
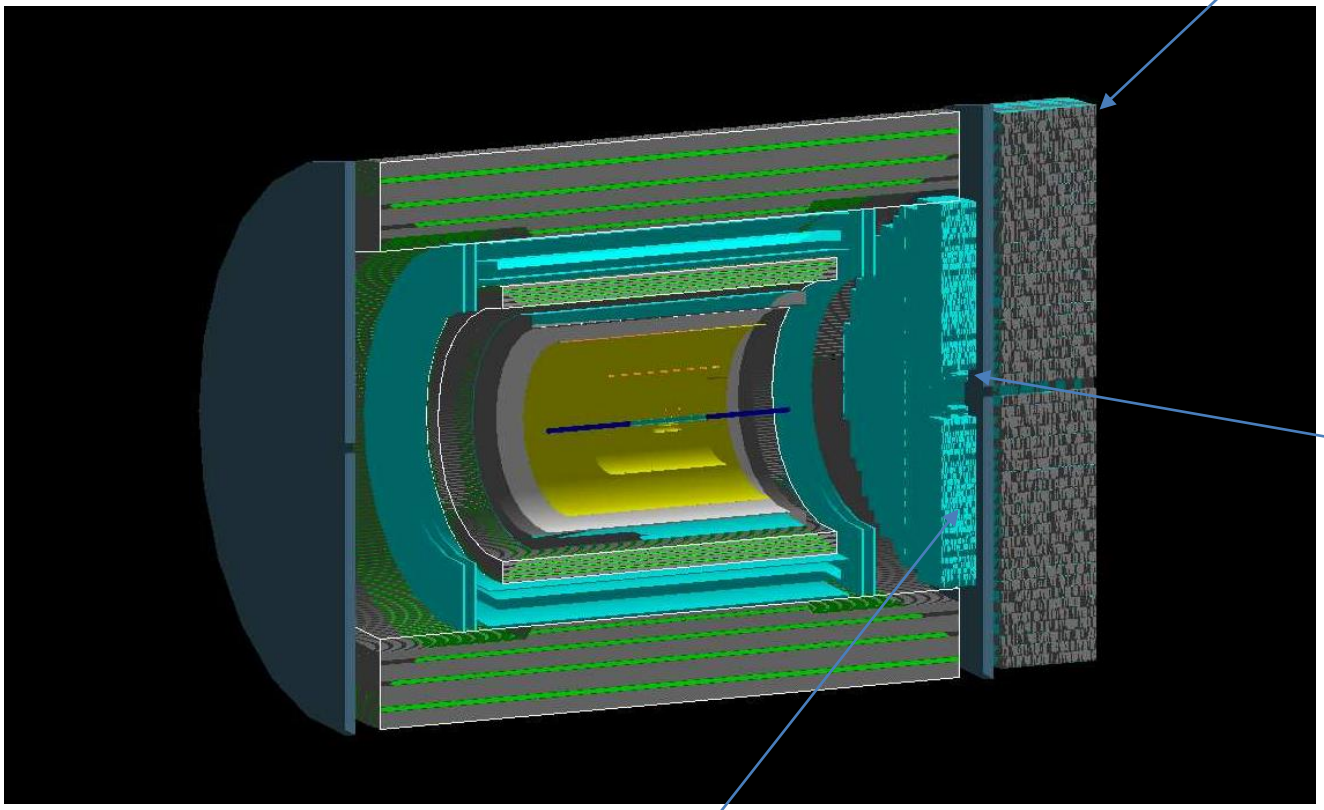


Figure A.2: The GEANT4 simulated jet resolution of single jets for energy (top row), ϕ (middle row) and η (bottom row) in $p+p$ (open markers) and $p+A$ (closed markers) collisions reconstructed with the FASTJET anti-k_T algorithm with $R = 0.4$ (blue) and $R = 0.6$ (red).

Backup Ken's slide in EICUG

Pb/Sc sandwich HCAL (NEW)
10 x 10 x 100 cm^3 towers ($1.2 < \eta < 4.0$)



20x20 array of
2.2 x 2.2 x 18 cm^3
PbW (PHENIX MPC) EMCal Crystals
with 10x10 square hole
(300 crystals total)
$3.0\text{-}3.3 < \eta < 4.0$

PHENIX PbSc EMCAL modules
(5.5 x 5.5 x 33 cm^3) organized in groups of four modules
(3152 modules or 788 groups of 4) ($1.4 < \eta < 3.0\text{-}3.3$)

Backup Pythia6: adding triggers (1)

```
bool PHPy6JetTrigger::Apply(const HepMC::GenEvent* evt)
{
    if (_verbosity > 2)
    {
        cout <<"PHPy6JetTrigger::Apply - HepMC event size: " <<evt->particles_size() <<endl;
    }

    // Loop over all particles in the event
    std::vector<fastjet::PseudoJet> pseudojets;

    int index = 0;
    for ( HepMC::GenEvent::particle_const_iterator p = evt->particles_begin();
          p != evt->particles_end();
          ++p )
    {
        index++;

        //Remove irrelevant particles:
        //12 = nu_e
        //13 = muons
        //14 = nu_mu
        //15 = taus
        //16 = nu_tau
        if ( (abs((*p)->pdg_id()) >= 12) && (abs((*p)->pdg_id()) <= 16) ) continue;

        //Avoid 0 pT events
        if ( ((*p)->momentum().px() == 0.) && ((*p)->momentum().py() == 0.) ) continue;

        //eta cut
        if ( ((*p)->momentum().pseudoRapidity() < _theEtaLow) ||
             ((*p)->momentum().pseudoRapidity() > _theEtaHigh) ) continue;

        //Energy cut
        if ( (*p)->momentum().e() < _minEnergy ) continue;

        //Save jets
        fastjet::PseudoJet pseudojet(
                (*p)->momentum().px(),
                (*p)->momentum().py(),
                (*p)->momentum().pz(),
                (*p)->momentum().e()
                );
        pseudojet.set_user_index(index);
        pseudojets.push_back(pseudojet);
    }
```

Backup Pythia6: adding triggers (2)

```
    //Call FastJet
    fastjet::JetDefinition *jetdef = new fastjet::JetDefinition(
            fastjet::antikt_algorithm, _R, fastjet::E_scheme, fastjet::Best
            );
    fastjet::ClusterSequence jetFinder(pseudojets, *jetdef);
    std::vector<fastjet::PseudoJet> fastjets = jetFinder.inclusive_jets();
    delete jetdef;

    bool jetFound = false;
    double max_pt = -1;
    for (unsigned int ijet = 0; ijet<fastjets.size(); ++ijet)
    {
        const double pt = sqrt(pow(fastjets[ijet].px(),2) + pow(fastjets[ijet].py(),2));

        if (pt > max_pt) max_pt = pt;
        if (pt > _minPt)
        {
            jetFound = true;
            break;
        }
    }

    if (_verbosity > 2)
    {
        cout <<"PHPy6JetTrigger::Apply - max_pt = " <<max_pt <<", and jetFound = " <<jetFound <<endl;
    }

    return jetFound;
}//Main
```

Backup Pytune message

```
pysubs.ckin[index-1] =value;
cout << "ckin\t" << index << " " << value << endl;
  }
else if ( label == "parp" )
  {
line >> index >> value;
pypars.parp[index-1] = value;
cout << "parp\t" << index << " " << value << endl;
  }
else if ( label == "parj" )
  {
line >> index >> value;
pydat1.parj[index-1] = value;
cout << "parj\t" << index << " " << value << endl;
  }
else if ( label == "paru" )
  {
line >> index >> value;
pydat1.paru[index-1] = value;
cout << "paru\t" << index << " " << value << endl;
  }
else if ( label == "parf" )
  {
line >> index >> value;
pydat2.parf[index-1] = value;
cout << "parf\t" << index << " " << value << endl;
  }
else if ( label == "pytune" ) //!
{
    line >> ivalue;
    pytune(&ivalue);
    cout << "pytune\t" << ivalue << endl;
}
else if ( label == "mdme" )
{
  int idc = 0;          // decay channel
  line >> idc >> index >> value;
```

```
root [0]
Processing Fun4All_G4_fsPHENIX.C...
Warning in <TClassTable::Add>: class PHG4ScintillatorSlat already in TClassTable
PHPythia6::read_config - Reading phpythia6.cfg
roots   200
proj    p
targ    p
frame   cms
 ************************************************************************
 *                                                                      *
 *   PYTUNE : Presets for underlying-event (and min-bias)               *
 *            Last Change : Aug 2013 - P. Skands                        *
 *                                                                      *
 *    100 Tune A                                                        *
 *            see R.D. Field, in hep-ph/0610012                         *
 *            and T. Sjostrand & M. v. Zijl, PRD36(1987)2019            *
 *                                                                      *
 *     MSTP(51) =            7   PDF set                                *
 *     MSTP(52) =            1   PDF set internal (=1) or pdflib (=2)   *
 *     MSTP( 3) =            2   INT switch for choice of LambdaQCD     *
 *     PARP(62) =       1.0000   ISR IR cutoff                          *
 *     PARP(64) =       1.0000   ISR renormalization scale prefactor    *
 *     PARP(67) =       4.0000   ISR Q2max factor                       *
 *     MSTP(68) =            3   ISR phase space choice & ME corrections*
 *            (Note: MSTP(68) is not explicitly (re-)set by PYTUNE)     *
 *     PARP(71) =       4.0000   IFSR Q2max factor in non-s-channel procs*
 *     PARJ(81) =       0.2900   FSR LambdaQCD (inside resonance decays)*
 *     PARJ(82) =       1.0000   FSR IR cutoff                          *
 *     MSTP(33) =            0   "K" switch for K-factor on/off & type  *
 *     MSTP(81) =            1   UE model                               *
 *     PARP(82) =       2.0000   UE IR cutoff at reference ecm          *
 *     PARP(89) =    1800.0000   UE IR cutoff reference ecm             *
 *     PARP(90) =       0.2500   UE IR cutoff ecm scaling power         *
 *     MSTP(82) =            4   UE hadron transverse mass distribution*
 *     PARP(83) =       0.5000   UE mass distribution parameter         *
 *     PARP(84) =       0.4000   UE mass distribution parameter         *
 *     PARP(87) =       0.7000   UE qq enhancement at low pT            *
 *     PARP(88) =       0.5000   UE qq enh scale / pT0                  *
 *     PARP(85) =       0.9000   UE gg color correlated fraction        *
 *     PARP(86) =       0.9500   UE total gg fraction                   *
 *     MSTP(91) =            1   BR primordial kT distribution          *
 *     PARP(91) =       1.0000   BR primordial kT width <|kT|>          *
 *     PARP(93) =       5.0000   BR primordial kT UV cutoff             *
 *    ------------------------------------------------------------------*
 *     MSTJ(11) =            4   HAD choice of fragmentation function(s)*
```